CLAIMS OF THE INVENTION

I claim:

1. A programmable graphics processor for executing a plurality of programs, said programmable graphics processor comprising:

an execution pipeline having a pipeline latency;

an interleaver for interleaving instructions from said plurality of programs and providing said instructions to said pipeline for execution such that each one of said programs has an apparent latency less than said pipeline latency.

2. The graphics processor of claim 1 wherein said pipeline has a datapath with a depth equal to said number of programs.

3. The graphics processor of claim 1 wherein a next instruction from one of said plurality of programs is not provided to said pipeline until a previous instruction of said one of said plurality of programs has completed.

4. The graphics processor of claim 1 wherein each program of said plurality of programs is independent of the other of said plurality of programs.

5. The graphics processor of claim 1 further including an output buffer for storing out of order data output.

6. The graphics processor of claim 1 further including one or more of a register copy, program counter, and program counter stack provided for each of said plurality of programs.

7. The graphics processor of claim 6 wherein one or more of control and computing resources, instructions, instruction memory, data paths, data memory, and caches are shared by said plurality of programs.

8. The graphics processor of claim 1 wherein said processor executes SIMD vector instructions of vector length N and executes in parallel a plurality of instructions having SIMD vector lengths that sum up to N.

9. The graphics processor of claim 1 wherein said instructions comprise load instructions for loading data from a data memory.

10. The graphics processor of claim 1 wherein said instructions comprise store instructions for storing data in a memory.

11. The graphics processor of claim 9 wherein said data memory comprises a cache.

12. The graphics processor of claim 9 wherein address space of said data memory comprises a frame buffer unit.

13. The graphics processor of claim 9 wherein address space of said data memory comprises a texture memory unit.

14. A method of executing instructions from a plurality of graphic processing programs comprising:

identifying N programs of said plurality of programs;

interleaving instructions from said N programs in a graphics processing execution pipeline;

executing said instructions such that a first instruction from any of said N programs is completed before beginning execution of a second of any of said N programs.

15. The method of claim 14 further including the step of assigning a program counter to each of said N programs.

16. The method of claim 14 further including the step of assigning a register to each of said N programs.

17. The method of claim 14 wherein said graphics processing execution pipeline has a depth of N.

18. The method of claim 14 wherein said processor executes SIMD vector instructions of vector length N and executes in parallel a plurality of instructions having SIMD vector lengths that sum up to N.

19. A method of executing a plurality of programs each having a plurality of instructions to be executed, comprising:

determining a desired output order of said plurality of programs,

assigning an output register slot to each of said plurality of programs,

loading an instruction from each of said programs in a graphics processing execution pipeline,

executing a loaded instruction;

loading program output into said assigned register slot.

20. The method of claim 19 further including loading a no-op in said execution pipeline for a program that has completed when remaining programs have instructions for execution.

21. The method of claim 19 further including loading new program when one of said register slots is available.

22. The method of claim 19 wherein a first instruction of any of said plurality of programs completes before a second instruction of any of said plurality of programs is provided for execution.